УДК 004.7

ПРОТИВОДЕЙСТВИЕ АТАКАМ НА ОТКАЗ В СЕТИ ИНТЕРНЕТ: ВЫБОР СРЕДЫ МОДЕЛИРОВАНИЯ

А.П. Игнатенко, Д.В. Цицкун

Институт программных систем НАН Украины,
63187, Киев, проспект Академика Глушкова, 40.
Тел.: 8(044)526 6025, E–mail: o.ignatenko@isofts.kiev.ua, scorpius1981@mail.ru.

В работе рассмотрена проблема противодействия атакам на отказ в сети Интернет. Рассмотрен этап построения модели сетевых взаимодействий. Проведен обзор существующих инструментов создания моделей и выполнен их сравнительный анализ.

This work deals with denial of service attacks problem. Stage of network model building is considered. Overview of modeling and simulating tools is provided.

Введение

Еще несколько десятилетий тому назад компьютерные системы были преимущественно однопользовательскими и обменивались информацией по нескольким довольно ограниченным каналам. Разработка нового принципа построение сети на основе коммутации пакетов позволили значительно повысить гибкость и живучесть системы. В настоящее время любая деятельность, связанная с обменом информацией не обходится без использования компьютерных сетей. Пропускная способность и охват самой крупной сети – сети Интернет постоянно возрастает. Это позволяет создавать многопользовательские, сильно распределенные приложения для работы по всему миру. Такие системы широко используются в областях кредитования, страхования, здравоохранения, права, военных приложений, связи и многих других.

Функционирование всех этих приложений существенно зависит от защищенности информационных потоков, поскольку эффективное совместное использование информационных ресурсов позволяют в значительной мере повысить качество обслуживания потребителей, продуктивность работы бизнеса, работы государственных служб. Таким образом, для организаций и отдельных пользователей характерна высокая степень связанности через открытые сети и зависимость функционирования от их бесперебойной работы.

Вместе с тем повсеместное внедрение сетей увеличило количество потенциальных злоумышленников, имеющих доступ к открытым системам. Одним из опасных видов преступной деятельности в сети Интернет являются так называемые атаки на отказ [1, 2]. Атака на отказ заключается в блокировании доступа к сервису, который предоставляется системой. Вообще говоря, существует два способа осуществления атаки. Первый способ использует уязвимости программного обеспечения, установленного на объекте атаки, что позволяет нападающим обрушить систему путем пересылки специально подобранных вредоносных пакетов. Второй способ заключается в использовании больших объемов бессмысленного трафика для исчерпания ресурсов жизненно важных элементов системы. Особенно опасными являются атаки направленные на узкие места сети – в этом случае работа всей системы может быть парализована.

Поэтому исследования по безопасности компьютерных систем актуальны как на стадии проектирования, так и при построении сетей. В этой связи важным элементом разработки сложной сетевой системы является средства имитационного моделирования [3]. Адекватно построенная модель позволяет эффективно анализировать уязвимости сети, оценить затрату времени и денег на ее построение и заранее просчитать основные характеристики ее функционирования. Безусловно, ключевым элементом при этом является построение модели, которая должна охватывать различные уровни работы сети, такие как физический, пакетный, пользовательский. Вследствие этого, модель часто содержит различные уровневые модели, представляющие работу сети в различных разрезах. При этом допускается использование как непрерывных, так и дискретных моделей. Например, потоки данных и скорость их передачи могут представляться непрерывными переменными, в то время как управляющие пакеты – дискретными.

В данной работе основное внимание будет сосредоточено на обзоре средств имитационного моделирования и применении к противодействию атакам на отказ.

Проблема защиты от атак типа «Распределенный отказ в обслуживании»

На сегодняшний день существует много различных видов атак на отказ, каждый из которых использует определенную особенность построения сети или уязвимости программного обеспечения. Например, атака может выполняться путем непосредственной пересылки большого количества пакетов (SYN, UDP, ICMP flood), использования эффекта отражения трафика (Smurf, Fraggle), передачи некорректных пакетов (Ping of Death, Land) или трудоемких запросов.

При этом с развитием методов противодействия возникают новые более сложные виды атак, к примеру атаки ухудшения качества (Quality Reduction Attack) и низкочастотные атаки (Low Rated Attack). Этот процесс безусловно будет продолжаться, требуя новых исследований и новых методов борьбы.

Отдельным видом атак на отказ являются распределенные атаки (Distributed Denial of Service Attacks), которые могут использовать тысячи взломанных «зомби»-хостов. Этими «зомби»-хостами неумышленно становятся миллионы незащищенных компьютеров, которые выходят в Интернет через широкополосные постоянно-активные соединения. Подсаживая на эти компьютеры «спящие» коды, хакеры получают возможность быстро создать группы агентов, действиями которых можно руководить удаленно. Если в атаке участвует достаточно большое количество «зомби»-хостов, ее масштабы могут быть поистине колоссальными.

Успешная атака на отказ вызывает самые разнообразные последствия: существенно ухудшается быстродействие сайтов, что вызывает обоснованное недовольство клиентов и других пользователей. Нарушаются обязательства по договорам обслуживания (Service-level agreements, SLA) и приходится возвращать немалые деньги за неоказанные услуги. На репутацию компаний ложится темное пятно, иной раз несмываемое. Неполученные прибыли, падение производительности, рост расходов на IT, судебные издержки – убытки расширяются и растут. По оценке Forrester, IDC, и прогнозам Yankee Group убытки от 24-часового перерыва в работе для крупной компании в сфере электронной коммерции приближаются к US$30 млн. Серия атак на Amazon, Yahoo, eBay и другие крупные сайты в феврале 2000 года, по оценкам Yankee Group, принесла совокупные убытки в размере US$1,2 млрд. А в январе 2001 года Microsoft из-за распределенной атаки на ее сайт потеряла около US$500 млн. всего за несколько дней.

Широкое описание различных видов атак на отказ приведено в [1, 4−8]. Ограничимся в этой работе несколькими существенными характеристиками. Целью атаки на отказ могут быть следующие элементы сети:

- отдельный сервис;
- отдельный узел;
- стационарная корпоративная сеть;
- динамическая (ad hoc) сеть;
- элементы инфраструктуры глобальной сети;

В работе [4] выделено множество аспектов атаки, важных с точки зрения построения системы защиты. Опишем вкратце эти аспекты:

Тип атаки. Будем рассматривать атаки двух типов: простую (когда атака идет с одной машины) и распределенную (когда используются машины-агенты).

Направление атаки – определяет конкретную часть инфраструктуры сети, которая подвергается. Как правило выделяют две части: ресурсы сети (т.е. пропускных каналов) и ресурсы цели (т.е. ресурсы конкретного компьютера) [1].

Схема атаки – определяет план осуществления атаки, т.е. доставки вредоносного трафика жертве. Может быть прямой (трафика отправляется с одной или нескольких машин), отраженной (трафика отражается через компьютеры третих лиц) или скрытой (вредоносный трафик скрыт внутри обычного), [6].

Способ атаки – определяет какие уязвимости используются при осуществлении атаки. Существуют следующие способы атаки: направленная, использующая недостатки конкретных приложений, служб, протоколов, поглощающая, которая пытается исчерпать все ресурсы системы или сети, эксплойтная, использующая уязвимости программных систем [8].

Требования к построению модели

При исследовании атак на отказ и проектировании системы защиты важно предварительно тестировать поведение системы защиты и сети в целом при проведении массированных распределенных атак. Эту работу следует проводить несколькими последовательными шагами, на различных уровнях детализации [9]:

- аппаратные стенды (натурное моделирование);
- эмуляция (полунатурное моделирование);
- имитационное моделирование на уровне пакетов;
- гибридное (аналитико-имитационное) моделирование;
- аналитическое моделирование.

В данной работе в основном внимание будет сосредоточено на трех последних уровнях. При этом собственно процесс моделирования состоит из пяти этапов:

- определение проблемы;
- построение модели;
- выполнение модели;
- анализ результатов;
- принятие решения об адекватности модели.

Ключевым решением при построении моделирующего средства является выбор среды для создания моделей, поскольку от этого будут зависеть возможности проведения экспериментов, представления результатов и доработки моделей.

Выбор среды для моделирования

На этапе построении модели возникает вопрос о выборе инструментария. При этом важными качествами для создания эффективной модели являются:

- детальная реализация протоколов, которые задействованы в атаках на отказ;
- возможность написания и подключения собственных модулей для реализации агентного подхода;
- возможность изменения параметров моделирования во время проведения экспериментов;
- платформенная независимость;
- развитый графический интерфейс;
- цена продукта;
- оценка ущерба атаки.

При исследовании предметной области авторами было выделено множество инструментов моделирования, наиболее подходящих для анализа сетевых процессов:

- **NS-2** − объектно-ориентированный программный продукт, ядро которого реализовано на языке C++. На базе ns2/nam возможна организация наглядной демонстрации функционирования протоколов и сетевых механизмов.
- **COMNET III** − объектно-ориентированная система моделирования локальных и глобальных сетей. Позволяет моделировать уровни: приложений, транспортный, сетевой, канальный. Использует все известные на сегодня технологии и протоколы, а также системы клиент-сервер. Легко настраивается на модель оборудования и технологий.
- **Netmaker** – проектирование топологии, средства планирования и анализа сетей широкого класса. Состоит из различных модулей для расчета, анализа, проектирования, визуализации, планирования и анализа результатов.
- **OPNET** − средство для проектирования и моделирования локальных и глобальных сетей, компьютерных систем, приложений и распределенных систем. Включает следующие продукты: Netbiz (проектирование и оптимизация вычислительной системы), Modeler (моделирование и анализ производительности сетей, компьютерных систем, приложений и распределенных систем), ITGuru (оценка производительности коммуникационных сетей и распределенных систем).
- **OMNeT++** представляет собой симулятор дискретных событий, которые происходят внутри простых модулей (simple modules). В системе OMNeT++ заложена детальная реализация протоколов, начиная от сетевого уровня, возможность написания и подключения собственных модулей, развитый графический интерфейс.

Приведем краткое описание каждого из этих продуктов.

NS-2, − объектно-ориентированный программный продукт, ядро которого реализовано на языке C++. Язык скриптов (сценариев) OTcl (Object oriented Tool Command Language) используется в качестве интерпретатора. ns2 полностью поддерживает иерархию классов C++ (называемую в терминах ns2 компилируемой иерархией) и подобную иерархию классов интерпретатора OTcl (называемую интерпретируемой иерархией). Обе иерархии обладают идентичной структурой, т.е. существует однозначное соответствие между классом одной иерархии и таким же классом другой.

Использование двух языков программирования в ns2 объясняется следующими причинами. С одной стороны, для детального моделирования протоколов необходимо использовать системный язык программирования, обеспечивающий высокую скорость выполнения и способный манипулировать достаточно большими объемами данных. С другой стороны, для удобства пользователя и быстроты реализации и модификации различных сценариев моделирования привлекательнее использовать язык программирования более высокого уровня абстракции. Такой подход является компромиссом между удобством использования и скоростью. В ns2 в качестве системного языка используется C++, позволяющий обеспечить:

- высокую производительность;
- работу с пакетам потока на низком уровне абстракции модели;
- модификацию ядра ns2 с целью поддержки новых функций и протоколов.

В качестве языка программирования высокого уровня абстракции используется язык скриптов OTcl, позволяющий обеспечить ряд положительных свойств, присущих языку Tcl/Tk (т.к. OTcl является объектно-ориентированным расширением языка Tcl):

- простота синтаксиса;
- простота построения сценария моделирования;
- возможность соединения воедино блоков, выполненных на системных языках программирования и простую манипуляцию ими.

Объединение для совместного функционирования C++ и OTcl производится с помощью TclCl (Classes Tcl). TclCl − интерфейс между объектами C++ и OTcl, которым пользуются ns2 и nam. Пример главного окна ns2 показан на рис. 1.

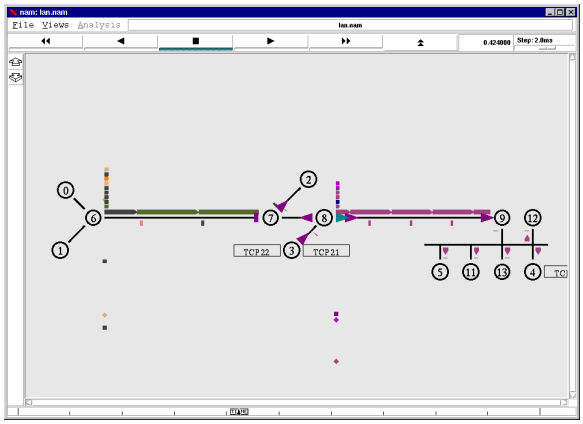

Рис. 1. Главное окно стимулятора ns2

Однако, архитектура ns-2 не оптимальна и требует доработок. В частности, разделяемая модель объектов C++/Otcl представляет некоторые проблемы:

1. Малая распространенность и документированность языка OTcl, а также сложность отладки написанных с его помощью скриптов. Во-вторых, существуют ограничения на совместное использование объектов C++. Это затрудняет создание новых комбинаций объектов, исходя из чего было решено отказать от разделяемой объектной модели в пользу реализованной полностью на C++ структуры, контролируемой с помощью скриптов.

2. Реализм. Базовая структура симулятора должна достаточно точно соответствовать структуре реальных сетей и их элементов. При проектировании нового объекта, функции или интерфейса необходимо отдавать предпочтение той структуре, которая имеет место в действительности. Этот принцип существенно влияет на переносимость кода и возможность использования симулятора в образовании.

3. Эффективность использования памяти. Симулятор должен поддерживать оба потока данных – полный и ограниченный. Другими словами, моделируемое приложение должно переслать 100,000 байтов данных удаленному узлу, но фактическое содержимое данных может быть достаточно абстрактным. На данный момент симулятор ns-2 оптимизирован для ограниченных потоков данных.

4. Трассировка. Для упрощения работы с моделями с большим количеством элементов и уменьшения объема регистрируемых и трассируемых событий, статистика моделирования должна гибко конфигурироваться пользователем. Некоторые возможности уже присутствуют в симуляторе ns-2: возможность трассирования только пакетов или слежение за определенным каналом. Более того, необходимо позволить пользователю определять тип регистрируемых данных, например, нужно ли регистрировать порты источника и приемника при исследовании протокола TCP.

5. Статистика. Упростить работу с симулятором позволит использование специальных объектов для сбора данных в процессе моделирования. Симулятор ns-2 поддерживает возможность интеграции таких объектов. Необходимо расширить набор объектов гистограммами, графиками последовательности событий, объектами для поиска максимальных и минимальных значений, распределения вероятности.

6. Топология. Необходимо определить набор базовых топологий (звезда, дерево, случайная топология с заданного размера) с переменными параметрами, что позволит упростить процесс создания моделей.

7. Визуализация. Симулятор должен поддерживать какую-либо форму анимации любой части модели, что применимо при отладке и демонстрации работы сети. Средство анимации nam должно входить в состав стимулятора.

Очевидно, что в настоящее время программный продукт ns2 является не оптимальным средством моделирования сетей связи. На базе ns2/nam возможна организация наглядной демонстрации функционирования протоколов и сетевых механизмов, например, влияния дисциплины обслуживания очереди на вероятность потери пакета трафика с разными приоритетами, или в чем заключается различие алгоритмов протокола TCP (slowstart, sliding window, SACK и т.д.).

COMNET. Семейство COMNET включает следующие системы:

COMNET III – система стохастического дискретного событийного моделирования систем массового обслуживания. Позволяет детально моделировать сети как СМО, построенные с использованием всех известных технологий и протоколов, как то: ATM, Frame Relay, FDDI, TCP/IP, клиент-сервер и т.д. Результатами моделирования являются оценки производительности различных вариантов построения исследуемой локальной или глобальной сети, учитывая при этом стоимостные характеристики.

Advanced Features Pack − данный пакет предоставляет дополнительные возможности пакету COMNET III для точного моделирования распределенного программного обеспечения клиент − серверных архитектур.

COMNET Predictor – система быстрого временного анализа. Предоставляет возможность быстро оценить производительность локальных и глобальных сетей. На основе импортированных данных по топологии, протоколам и трафику пользователю предоставляется возможность изменить такие параметры, как топология, трафик, состав оборудования, полоса пропускания, протоколы и быстро получить результат в виде отчетных графических форм.

COMNET Baseliner − система импорта данных. Предназначен для импорта данных о топологии и протоколам из установленных у пользователя систем управления и мониторинга сетей с целью создания базовых моделей для пакетов COMNET III и COMNET Predictor.

COMNET Enterprise Profiler – система мониторирования сети. Позволяет производить мониторирование и сбор статистики в сети без возможности администрирования. Может интегрироваться с другими системами мониторирования и управления.

Network II.5 − автономный пакет для анализа производительности используемых компьютерных систем. Позволяет проводить моделирование компьютерной архитектуры любого типа.

Общий вид главного окна системы показан на рис. 2.

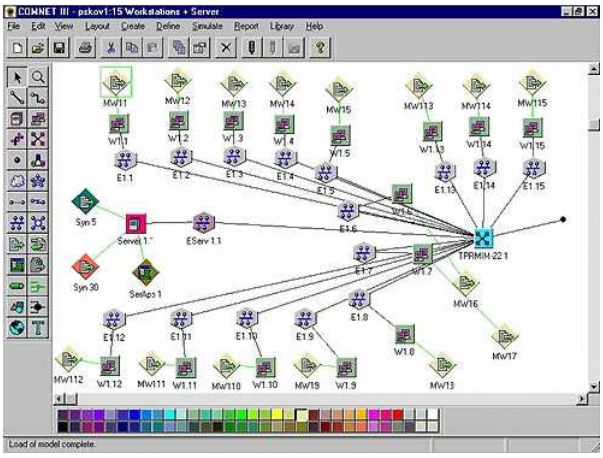

Рис. 2. Главное окно системы

При моделировании в COMNET затрагиваются следующие уровни эталонной модели взаимодействия открытых систем (OSI ISO + IEEE 802): приложений, транспортный, сетевой, канальный. На уровне приложений описываются источники трафика − сообщения, сеансы, отклики, вызовы, поведение программного обеспечения. На транспортном уровне − транспортные протоколы и их параметры. На сетевом уровне: алгоритмы маршрутизации, потоки пакетов, таблицы маршрутизации штрафные функции. Канальный уровень - непосредственно передача пакетов, ретрансляция, описание каналов.

OPNET. Opnet Modeler предлагает пользователям графическую среду для создания, выполнения и анализа событийного моделирования сетей связи. Это удобное программное обеспечение может быть использовано для большого ряда задач, например, типичные создание и проверка протокола связи, анализ взаимодействий протокола, оптимизация и планирование сети. Также возможно осуществить с помощью пакета проверку правильности аналитических моделей, и описание протоколов.

Общий вид главного окна системы показан на рис. 3.

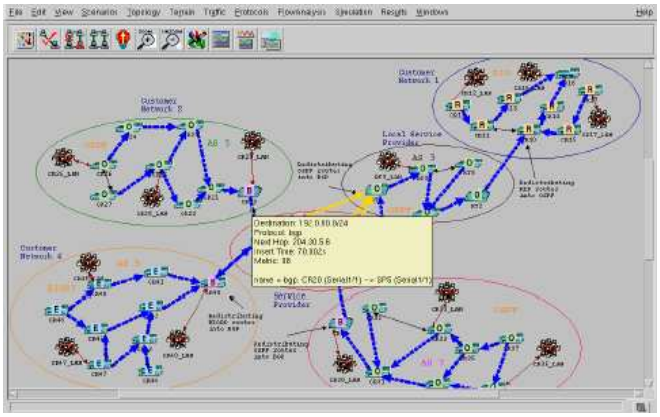

Рис. 3. Общий вид главного окна системы Opnet

В рамках, так называемого, редактора проекта могут быть созданы палитры сетевых объектов, которым пользователь может присвоить различные формы соединения узлов и связи вплоть до имеющих вид головоломки. Автоматизированное порождение сетевой топологии − кольца, звезды, случайной сети, также поддерживается и резервируется утилитами для импортируемых сетевых топологий в различных форматах. Случайный трафик может быть автоматически сгенерирован из алгоритмов, указанных пользователем, а также импортирован из входящих в стандартную комплектацию пакета форматов реальных трафиков линий. Результаты моделирования могут быть проанализированы, а графы и анимация трафика, опять же будут сгенерированы автоматически. Новая особенность − это автоматическое преобразование в формат html 4.0x.

Одним из плюсов из создания модели сети с помощью программного обеспечения является то, что уровень гибкости, обеспечиваемый ядром моделирования, тот же, что и для моделирования, написанных с нуля, но объектное построение среды позволяет пользователю намного быстрее делать разработку, усовершенствования и производить модели для многократного использования.

Есть несколько сред редактора − по одной для каждого типа объекта. Организация объектов − иерархическая, сетевые объекты (модели) связаны набором узлов и объектов связи, при этом как объекты узла связаны набором объектов, типа модулей очерёдности, модулей процессора, передатчиков и приемников. Версия ПО для моделирования радиоканала содержит модели антенны радиопередатчика, антенны приемника, перемещающихся объектов узла (включая спутники).

Логику поведения процессора и модулей очередности определяет модель процесса, которую пользователь может создавать и изменять в пределах редактора процесса, в котором пользователь может определить модель процесса через комбинацию алгоритма работы конечного автомата (finite-state machine - FSM) и операторов языка программирования C/C++.

Вызов события модели процесса в течение моделирования управляется возбуждением прерывания, а каждое прерывание соответствует событию, которое должно быть обработано моделью процесса.

Основа связи между процессами − структура данных, называемая пакетом. Могут быть заданы форматы пакета, т. е. они определяют, какие поля могут содержать такие стандартные типы данных, как целые числа, числа с плавающей запятой и указатели на пакеты (эта последняя способность позволяют инкапсулировать моделирование пакета). Структура данных, вызывающая информацию по контролю за интерфейсом (interface control infor-mation - ICI), может быть разделена между двумя событиями моделей процесса − это ещё один механизм для межпроцессорной связи, это очень удобно для команд моделирования и соответствует архитектуре многоуровневого протокола. Процесс также может динамически порождать дочерние процессы, которые упростят функциональное описание таких систем, как серверы.

Несколько основных моделей процесса входят в базовую комплектацию пакета, моделируя популярные протоколы работы с сетями и алгоритмы, вроде протокола шлюза границы (border gateway protocol - BGP), протокола контроля передачи. Интернет протокол (TCP/IP), ретрансляции кадров (frame relay), Ethernet, асинхронного режима передачи (asynchronous transfer mode -ATM), и WFQ (weighted fair queuing). Базовые модели полезны для быстрого развития сложных имитационных моделей для общих архитектур сети, а также для обучения, чтобы дать точное функциональное описание протокола студентам. Существует возможность сопровождения комментариями и графикой (с поддержкой гипертекста) моделей сети, узла или процесса.

Одним из недостатков системы является большая стоимость на представленный продукт.

NetMaker XA. Вычислительное ядро моделирования, используемое в NetMaker XA от Make Systems, − одно из наиболее мощных на рынке, и это сыграло немаловажную роль в том, что продукт зарекомендовал себя столь хорошо. Все работает в полном соответствии с описаниями. У разработчиков не возникает никаких проблем ни с моделированием небольшой сети, ни с усовершенствованием системы, приведенной производителем в качестве примера. Кроме того, генерируемые программой отчеты содержали всю необходимую информацию. Главные недостатки NetMaker XA − необходимость серьезного обучения пользователя и высокая стоимость. Если к цене базовой конфигурации изделия добавить стоимость дополнительных модулей, получится довольно значительная сумма.

Основу продукта составляют модули Visualizer, Planner и Designer. Каждый из них выполняет какую-то одну функцию; чтобы смоделировать работу сети, необходимы все три. Visualizer служит для получения информации о сети и ее просмотра. В его состав входят SNMP-модули автоматического распознавания, которые опрашивают сетевые устройства и создают соответствующие им объекты. Информацию об этих объектах можно затем редактировать с помощью Visualizer.

Planner − это библиотека устройств, которая помогает проанализировать, что получится при установке в сети нового устройства (например, дополнительного маршрутизатора). Make Systems поставляет встраиваемые модули (plug-in), содержащие объекты с данными о продуктах различных производителей. В таких объектах содержится полное описание различных моделей устройств (от числа сетевых интерфейсов до типа процессора); вся информация заверяется производителем. С помощью Planner пользователь может самостоятельно строить свои собственные объекты для описания сетевых устройств и каналов связи, не включенных в библиотеку. Designer нужен для построения схем сетей. Данное средство позволяет легко и быстро создавать модели и анализировать альтернативы. Если пользоваться им совместно с Planner, можно получать информацию о том, как будет работать сеть заданной конфигурации.

Если требуется пойти несколько дальше, придется приобрести еще три модуля: Accountant, Interpreter и Analyzer. В состав Account входит тарификационная база данных; этот модуль помогает проанализировать затраты, связанные с использованием тех или иных сетей общего доступа. Модуль Interpreter предназначен для сбора данных от средств анализа трафика. Система построена таким образом, что данные автоматически импортируются в созданную модель, что позволяет использовать их почти в режиме реального времени, а не строить гипотезы относительно работы сети. Наконец, Analyzer и предназначенный для него встраиваемый модуль "выживаемости" помогают разрабатывать планы восстановления после аварий, а также добиваться того, чтобы ни одна неисправность (после ее локализации) не могла привести к отказу сети в целом.

Стоит все это очень дорого − от 7 тыс. дол. за базовый комплект плюс доплаты за встраиваемые модули. Установить NetMaker XA можно только на SPARCstation от Sun Microsystems. В Make Systems осознают, что пользоваться их продуктом не так-то просто.

OMNeT++. Система OMNeT++ представляет собой симулятор дискретных событий, которые происходят внутри простых модулей (simple modules). Обмен сообщениями между модулями осуществляется по каналам (модули соединены с ними шлюзами) или непосредственно через шлюзы.

На основном окне визуализации (рис. 4, справа внизу) отображается компьютерная сеть для проведения моделирования. Она представляет собой набор клиентских станций, сервер и маршрутизатор, соединенных каналами связи. Каждый узел может нести различную функциональность в зависимости от параметров или набора внутренних модулей. Внутренние модули отвечают за работу протоколов и приложений на различных уровнях модели OSI. Узлы сети соединяются между собой каналами связи, параметры которых можно изменять. На основной окне визуализации (рис. 4, справа и слева сверху) отображается внутренняя структура сервера и маршрутизатора (протоколы сетевого, транспортного уровней модели OSI, таблица маршрутизации и т.д.). На рис. 4 в нижнем левом углу указано запись лога всех происходящих процессов. Заметим, что все вышеупомянутые свойства доступны в любое время протекания процесса. Тем самым, пользователь может отследить передвижение интересующего пакета по сети.

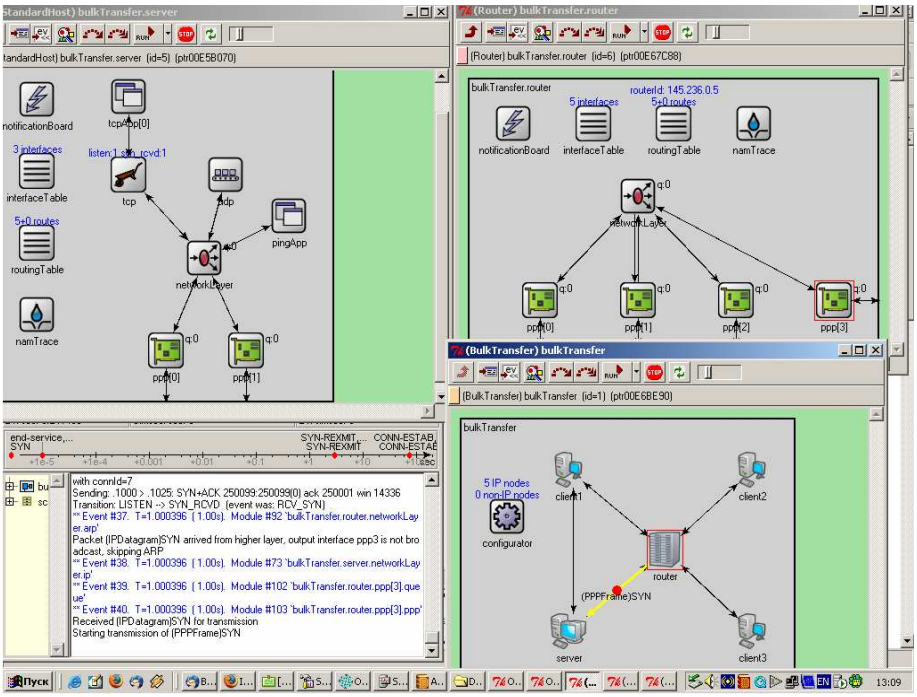

Рис. 4. Пример пользовательского интерфейса среды моделирования OMNeT++

Для описания топологии сети используется язык NED Language, который только определяет структуру и топологию сети, но оставляет без изменений поведение и значения подмножества параметров модуля.

Сравнительная характеристика

Для выбора инструментария моделирования сети и процессов передачи информации был проведен анализ целого ряда пакетов моделирования (имитаторов сетей).

По результатам обзора инструментария моделирования сети, для наглядности функциональных возможностей каждой из систем моделирования была создана таблица сравнений. На основании выводов по данным выбрано одну из систем моделирования компьютерных сетей.

И так, можно сделать следующий вывод, было выдвинуто ряд требований, которые предъявлялись к используемому имитатору, в частности, детальная реализация протоколов, начиная от сетевого уровня (для возможности моделирования основных классов сетевых атак), возможность написания и подключения собственных модулей для реализации агентского подхода, развитый графический интерфейс и др. Было выявлено, что этим требованиям в наибольшей степени удовлетворяет OMNeT++ INET Framework.

Таблица

Компания	Продукт	Построение схемы сети			Анализ			Стоимость	
		Наличие библиотеки	Настройка библиотеки устройств	Импорт информации о работе сети	Полный анализ ситуации	Пошаговая трассировки	Генерация отчетов	Базовая	Дополнительные модули
CACI Products, www.caciasl.com	COMNET III	+	+	+	+	+	+	20 000	5000-9000
Make Systems, www.makesystems.com	NetMaker XA	+	+	+	+	–	+	7 000	7000--15 000
MIL 3, www.mil3.com	OPNET Planner	+	–	+	+	–	–	17 000	9000--24 000
OMNEST team, www.omnest.com	OMNET++	+	+	+	+	+	+	–	–

1. *Уланов А. В., Котенко И. В.,* Защита от DDoS-атак: механизмы предупреждения, обнаружения, отслеживания источника и противодействия // Защита информации.− 2007. INSIDE, № 1-3.
2. *Mircovic J., Dietrich S., Dietrich D., Reiher P.* Internet Denial of Service: Attack and Defense // Mechanisms. Prentice Hall, Engle Wood Cliffs. 2005. N J.
3. *Bertsekas D. P.,* Network Optimization: Continuous and Discrete Models. Athena Scientific.–1998. – 270 p.
4. *Ігнатенко О.П.,* Виявлення низькочастотних атак на відмову на основі історичних даних // Комп'ютерні науки та інформаційні технології.−Львів; 2008.− № 1. – С.23-27.
5. *Xiang Y., Zhou W., Chowdhury M.,* A Survey of Active and Passive Defence Mechanisms against DDoS Attacks. Technical Report, TR C04/02, School of Information Technology, Deakin University, Australia.−March 2004.
6. *Specht S. and Lee R.,* Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures // Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems, 2004 International Workshop on Security in Parallel and Distributed Systems.–September 2004. – P. 543–550.
7. *Peng T., Leckie C., and Ramamohanarao K.*. Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems // ACM Computing Surveys. – April 2007. – Vol. 39, N 1. – P. 31–42.
8. *Rocky K., Chang C.,* Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial // IEEE Communications Magazine, October 2002. – P. 42 – 51.
9. *Negenborn R.R., Schutter B., and Hellendoorn H.,* "Multi-agent model predictive control for transportation networks with continuous and discrete elements," Proceedings of the 11th IFAC Symposium on Control in Transportation Systems, Delft, The Netherlands.–2006. – P. 609-614.